Optimal Route Planning for Probabilistic Landscape Exploration

Ming Tony Shing
Department of Physics
Hong Kong University of Science and Technology
Hong Kong
mtshing@ust.hk

K. Y. Michael Wong
Department of Physics
Hong Kong University of Science and Technology
Hong Kong
phkywong@ust.hk

***Abstract*—In this paper we formulate a new algorithm for planning optimal exploratory routes that maximize the total probability of finding resources, whose distribution is given by a probabilistic landscape. The problem resembles the Traveling Salesman Problem on a probabilistic map without well-defined routes or cities. Our algorithm incorporates the existing Elastic Net Algorithm with barrier method, which is able to produce optimal routes subject to total length constraints. We have tested the algorithm and found that its performance depends on symmetries of the landscape, separation constraints and search range over the landscape. Using the population density of North-East United States as a test bed, the algorithm produces excellent results.**

***Index Terms*—Route Planning, Optimization, Probabilistic Landscape, Surface Exploration**

I. Introduction

The optimal path planning is an important problem in the operation of mobile agents in navigation and resources collection. In a realistic scenario, the agents are sent out from base stations, navigate and collect resources on the map, and need to return to the base for recharging and unloading the gathered resources within a limited time. In planning such missions, says, for space exploration, it is common to manually set a goal point in the map and compute the optimal traverse between the base and the goal with path searching algorithms [1], [2]. In order to maximize the efficiency of the excursion, it would be ideal if we can plan a route which passes through many regions with densely distributed resources and with a return path which is different from the outgoing path, so that the agent can navigate more area with the same amount of excursion time. This problem is equivalent to a selective travelling salesman problem (TSP) on a weighted map [3], which in our case maximizes the score of a closed path under certain length constraints. However, TSP is a well-known NP-hard problem whose complexity increases exponentially with the number of nodes in the graph, while a weighted map is like a graph with millions of nodes, depending on its resolution. This makes exhaustive search unfeasible.

There has been work on the adaptation of TSP to continuous map, which aims to compute the optimal curves connecting the given cities [4]. However, the cities must be defined on the map before optimization. This requires manual decision of mining sites in the exploration missions, which excludes the possibility that the optimal path may not pass through the selected sites. On the other hand, common path searching algorithms such as A* [5] and D* [6] highly depend on the choices of the goals and the agent is not guaranteed to be able to collect optimal amount of resources along the direction of the manually selected goal. In practice, the distribution of resources is often probabilistic and the waypoints and goals are often not well-defined.

There have been many heuristic algorithms proposed for TSP, but most of them are graph-based and difficult to be modified for our need. The Elastic net algorithm (ENA) is a heuristic solution to the original TSP [7]. The search of ENA involves treating the distribution of nodes as discrete 'pits' on a flat plane and optimizing the fitting of Gaussian functions onto the pits. It is possible to extend its application to the searching on a weighted map which is like a landscape with continuous variations. Besides, ENA has an advantage that the number of variables is linear relative to the number of nodes in the path. However, although ENA provides a possible solution to our problem, it suffers from serious limitation in yielding uneven step lengths due to the lack of well-defined destinations as in the conventional TSP.

In this paper, we first formulate our problem as a selective TSP on continuous map and introduce an alternative to ENA using with logarithmic barrier functions as a heuristic solution, which can generate solutions with better practical use than using the original ENA. Details are described in section 2. Then we present some computational experiments of the algorithm to demonstrate the effectiveness of the algorithm and study how the results depend on the choice of parameters in section 3. Finally conclusions and outlook are presented in section 4.

II. Model Statement

We would like to construct an algorithm which yields practical solutions for mission planning. We formulate the problem as a rover running on a surface with a probablistic resource distribution. The rover has an effective search range as a circle of radius of K. The collection of resources is performed over a 2-dimension Euclidean space, and let $P(\vec{x})$ be the resource abundance at position $\vec{x}$ which is an arbitrary scalar function.

We aim to plan a closed traverse which allows the rover to maximally visit the locations of high resource abundance. The traverse is defined by a set of N nodes $\{\vec{y}_1, \vec{y}_2, \dots \vec{y}_N\}$. To evaluate the performance of the traverse, we define the score of the path as

$$S_K = \sum_{j=1}^{N} \log \int P(\vec{x}) \exp\left(-\frac{\|\vec{x}-\vec{y}_j\|^2}{2K^2}\right) d\vec{x}. \quad (1)$$

This is equivalent to applying a Gaussian filter around each node of the traverse and optimize by matching the geometrical structures on the map. We also define a "raw score" of the traverse to be

$$S_0 = \sum_{j=1}^{N} P(\vec{y}_j) \quad (2)$$

which is useful for comparing the performance when different values of K are used.

Furthermore, we suggest that the results would be more applicable by imposing separation constraints between the nodes. In traverse planning, the nodes on the traverse are often referred to as waypoints, which are time indicators to the schedule of the mission. We should impose lower and upper bounds $[d_{\min}, d_{\max}]$ to the separations between nodes, which depends on the velocity range of the rover, so that the travelling between waypoints within a given time interval is feasible.

We summarize the formulation of the problem as

$$\begin{aligned} \underset{\{\vec{y}_j\}}{\text{maximize}} \quad & S_K = \sum_{j=1}^{N} \log \int P(\vec{x}) \exp\left(-\frac{\|\vec{x}-\vec{y}_j\|^2}{2K^2}\right) d\vec{x} \\ \text{subject to} \quad & \|\vec{y}_j - \vec{y}_{j+1}\| \le d_{\max}, \text{ with } \vec{y}_{N+1} = \vec{y}_1 \\ & \|\vec{y}_i - \vec{y}_j\| \ge d_{\min}, \ i \ne j \\ & \text{for } i, j = 1, \dots N \end{aligned}$$

III. Methodology

A. Adaptation to Probabilistic Map

We first explored the possibility using the original ENA. ENA minimizes an energy equals to

$$\begin{aligned} E_{EN}^{\{\vec{x}_i\}} = & -\alpha K \sum_i \log \sum_j \exp\left(-\frac{\|\vec{x}_i - \vec{y}_j\|^2}{2K^2}\right) \\ & + \beta \sum_j \|\vec{y}_j - \vec{y}_{j+1}\|^2 \end{aligned} \quad (3)$$

where $\vec{x}_i$ represents the location of city i and $\vec{y}_j$ represents the j-th node on the elastic net, with N nodes on the net. The first term of the energy corresponds to the attraction between nodes on the net and cities in the neighbourhood; the second term imposes a quadratic tension between the nodes and tends to yield equal separation between nodes. Applying gradient descent method to the energy gives the update rule

$$\Delta\vec{y}_j^{\{\vec{x}_i\}} = \alpha \sum_i \omega_{ij}(\vec{x}_i - \vec{y}_j) + \beta K(\vec{y}_{j+1} + \vec{y}_{j-1} - 2\vec{y}_j) \quad (4)$$

where

$$\omega_{ij} = \frac{\exp\left(-\frac{1}{2K^2}\|\vec{x}_i - \vec{y}_j\|^2\right)}{\sum_j \exp\left(-\frac{1}{2K^2}\|\vec{x}_i - \vec{y}_j\|^2\right)}.$$

α, β are constants, and K is the scale parameter which is gradually reduced during iterations. To make an adaptation to the map of an arbitrary scalar function $P(\vec{x})$, we replace the summation of cities by an integration on the map, and modify the energy to

$$\begin{aligned} E_{EN}^{P(\vec{x})} = & -\alpha K \sum_j \log \int P(\vec{x}) \exp\left(-\frac{\|\vec{x}-\vec{y}_j\|^2}{2K^2}\right) d\vec{x} \\ & + \beta \sum_j \|\vec{y}_j - \vec{y}_{j+1}\|^2 \end{aligned} \quad (5)$$

with the update rule modified into

$$\Delta\vec{y}_j^{P(\vec{x})} = \alpha\vec{\Omega}_j + \beta K(\vec{y}_{j+1} + \vec{y}_{j-1} - 2\vec{y}_j) \quad (6)$$

where

$$\vec{\Omega}_j = \frac{1}{K^2} \frac{\int P(\vec{x}) \exp\left(-\frac{1}{2K^2}\|\vec{x}-\vec{y}_j\|^2\right)(\vec{x}_i - \vec{y}_j) dx}{\int P(\vec{x}) \exp\left(-\frac{1}{2K^2}\|\vec{x}-\vec{y}_j\|^2\right) dx}.$$

The first term in (5) is the same as the score of the path we defined in (1). However, we find that the algorithm is problematic - the resultant path fails to maintain the equal separation property as in the original TSP and the separation constraints are heavily violated. For example in Fig. 1, all nodes draw themselves to the peaks indicated by dark regions in the map and overlap with each other instead of staying apart as intermediate points along the path. Such paths suggest that we should do intensive search in the region with the highest resource density and ignore any locations in the middle. Yet this kind of missions is unpractical because it may be unfeasible for mobile agents with limited speed to travel across big leaps between neighbouring waypoints. Besides, the overlapping of the probabilities among the neighboring nodes implies that the total probability of finding resources is unnecessarily reduced.

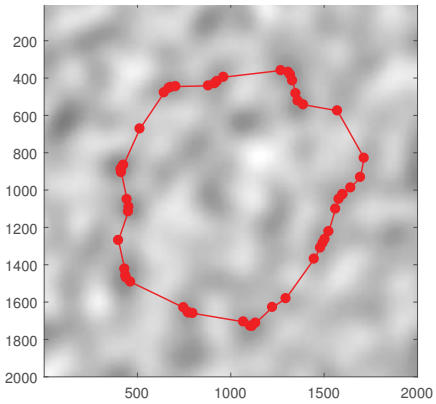


Fig. 1. Examples of a path generated by the original ENA on a random map. Darker colors represent regions with higher resource abundance. Nodes are strongly drawn to peaks on the map instead of spreading evenly as intermediate points. Thus the separation constraints are broken.

B. Constraints by Logarithmic Barrier

To produce paths for practical missions, we impose the separation constraints by logarithmic barrier functions [8]. This method is effective in finding unique optimal solutions for convex problems. However, as we shall see, the method may converge to multiple solutions as our path exploring problem is non-convex. For a minimization problem subject to an inequality constraint $f(x) \leq 0$, the method adds an additional logarithmic function $\phi(x)$ to the objective function

$$\phi(x) = -\beta \log(-f(x))$$

where β is a parameter gradually reduced during iterations. In our problem, we replace the elastic energy in the modified elastic net with two logarithmic functions which correspond to the upper and lower bound of the separation between nodes. The energy for minimization becomes

$$\begin{aligned} E_{EN}^{P(\vec{x})} = &- \alpha K \sum_j \log \int P(\vec{x}) \exp\left(-\frac{\|\vec{x}-\vec{y}_j\|^2}{2K^2}\right) d\vec{x} \\ &- \beta \sum_j \log\left(d_{\max} - \|\vec{y}_j - \vec{y}_{j+1}\|\right) \\ &- \gamma \sum_{i\neq j} \sum_j \log\left(\|\vec{y}_i - \vec{y}_j\| - d_{\min}\right). \end{aligned} \tag{7}$$

β, γ are parameters that will be gradually reduced during iteration. The β term ensures that the separation between each pair of successive nodes to is bounded below a maximum. The γ term ensures that the separation between all pairs of nodes are greater than a minimum because we also want to prevent non-neighbouring nodes overlapping each other in overcrowding regions. Again, gradient descent is used and we obtain the update rule as

$$\Delta \vec{y}_j^{P(\vec{s})} = \alpha \vec{\Omega}_j + \beta(\vec{T}_{j-1,j} - \vec{T}_{j,j+1}) + \gamma \sum_{i\neq j} \vec{R}_{i,j} \tag{8}$$

where

$$\vec{T}_{j-1,j} = \frac{1}{d_{\max} - \|\vec{y}_{j-1} - \vec{y}_j\|} \cdot \frac{\vec{y}_{j-1} - \vec{y}_j}{\|\vec{y}_{j-1} - \vec{y}_j\|}$$

$$\vec{R}_{i,j} = \frac{1}{d_{\min} - \|\vec{y}_i - \vec{y}_j\|} \cdot \frac{\vec{y}_i - \vec{y}_j}{\|\vec{y}_i - \vec{y}_j\|}.$$

With the introduction of barrier functions, the nodes on the resultant path are more evenly distributed, as shown in Fig. 2, which is more ready to be implemented in practical missions.

IV. Computational Experiments

We continue to test the performance of our algorithm on different maps by MATLAB. Each map is a gray-scale picture with every pixel in the value range $0 \leq P \leq 1$. We also implemented the method of backtracking step sizes [9] to speed up the computation. From the experimental results, we noticed that the performance by our algorithm mainly depends on 3 factors - (1) nodes' initial position, (2) restriction from separation constraints $[d_{\min}, d_{\max}]$ and (3) search range K.

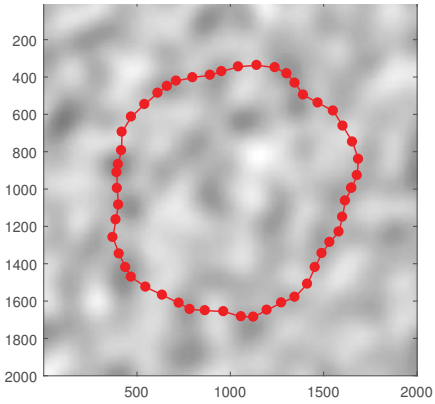


Fig. 2. Examples of a path generated by our algorithm with barrier functions, on the same random map as in Fig.1. With the help of barrier functions, the constraints can be maintained and the nodes are more evenly distributed.

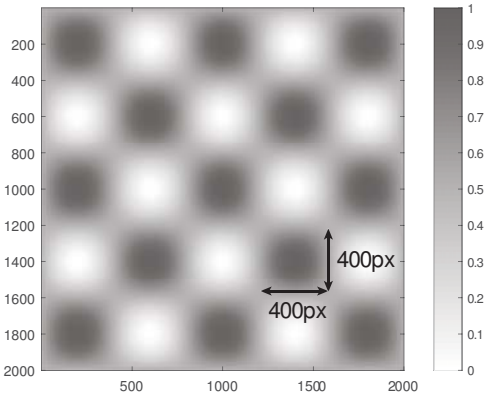
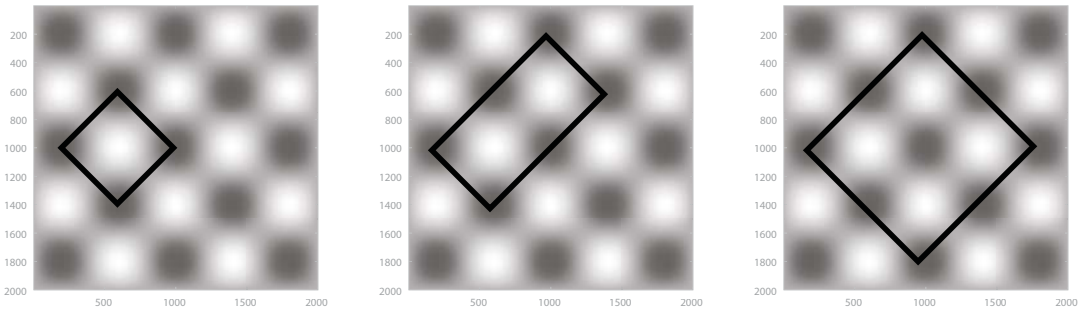


Fig. 3. The "grid-like" map for the following demonstrations. The darker regions represent regions with higher "resource abundance". The map has a size of $2000 \times 2000\text{px}^2$ and a correlation length of 400px.

A. Initial Position Dependency

The initial position dependency is demonstrated through a "grid-liked" map, which is generated by the function $P(x_1, x_2) = \frac{1}{2}\sin(\frac{\pi x_1}{400})\sin(\frac{\pi x_2}{400}) + \frac{1}{2}$, as shown in Fig. 3. This map is chosen because the optimal paths are intuitive - we can expect the optimal solutions are any paths that connect the peaks and snap on the "grid" lines, as one of the shapes in Fig. 4, or their variants by folding the corners. Among the 3 shapes of optimal paths, the "2x2" square is expected to be the global optimal because the path passes through 8 regions of high resource abundance, while "1x1" and "1x2" only passes through 4 and 6 peaks respectively.

We try to generate results which fit the shapes of any of the above. The simulations start with the initial node positions as a circle of varying radius R. We position the center of the initial circle at $(x, y) = (1000, 1000)$ and $(840, 920)$ and observe the effect of the initial positions. A path of 50 nodes with separation constraints $[d_{\min}, d_{\max}] = [10, 100]$ and search range $K = 50$ is used in all the cases. In this case all "1x1", "1x2", "2x2" are feasible since the maximum length of the path is $50 \times 100 = 5000$px.



(a) "1x1" (b) "1x2" (c) "2x2"

Fig. 4. Shapes of the three basic types of expected optimal paths. The shapes have a total length of 2262px, 3394px and 4525px respectively. "2x2" is expected to be the global optimal because it passes most regions of high resource density.

1) Initial Circle Center at $(x, y) = (1000, 1000)$*:* Due to the map's symmetry, the attraction forces to the circle from 4 sides are equal. As a result the algorithm is unable to generate optimal path that is asymmetric about the center of the map. Only "2x2" and "cross" (which is a variant of "2x2" by folding the corners to center) can be produced, as shown in Fig. 5. We also note that there is a transition from "cross" to "2x2" when the initial radius is is larger than the correlation length of the map (400px). It is because if the nodes are initially less than 400px away from the center, the inward attraction is stronger and so the nodes are pulled to the center region, resulting in a cross shape; If the nodes are more than 400px away from the center, outward attraction is stronger thus results in the "2x2" square.

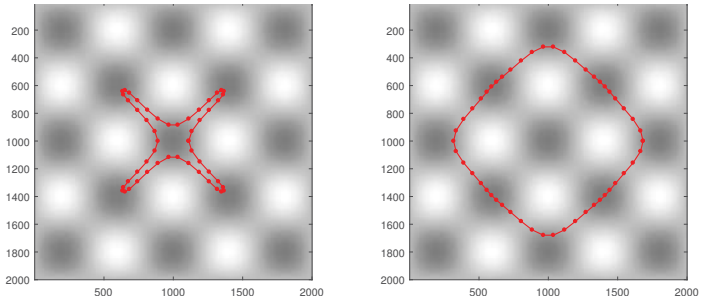
2) Initial Circle Center at $(x, y) = (840, 920)$*:* The center of the initial circle is chosen to deviate from the map's center so that the symmetry effect is broken. We observe all "1x1", "1x2", "2x2" and their variants while varying the radius of the initial circle, as shown in Fig. 6. Similar to the $(x, y) = (1000, 1000)$ case, transitions between shapes are due to the competition between inward and outward attraction. In general, we could expect a transition may occur when the initial radius is near a multiple of 200px (half side length) or $200\sqrt{2}$px (half diagonal length). For example, a transition from "L" shape to "1x1" occurs when the initial radius exceeds $200\sqrt{2} \approx 282$px, and transition from "1x1" to "1x2" occurs when the initial radius exceeds ≈ 400px. In deciding the initial position of the nodes, we can try multi-start with initial radii of multiples of the map's correlation length to obtain most of the possible configurations.

B. Length Restriction from Separation Constraints

We further carry out demonstrations with the "grid"-like map but impose stricter separation constraints $[d_{\min}, d_{\max}] = [10, 70]$. The maximum length of the path is $70 \times 50 = 3500$px, and so the "2x2" square becomes an unfeasible solution due to its total length being 4525px. We run the same two series of experiments, with the initial center located at $(1000, 1000)$ and $(840, 920)$.

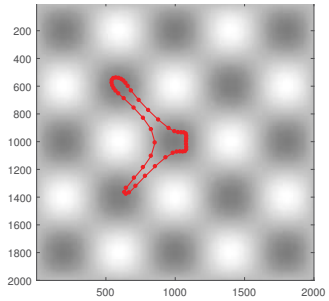
1) Initial Circle Center at $(x, y) = (1000, 1000)$ *with Length Restriction:* The results under this setting are shown in Fig. 7. Only a diminished "Cross" shape can be produced when we started with an initial circle of small radius. However, it is obviously worse than the result in Fig. 5a since its 4 arms cannot reach the darker regions. When a larger initial radius is used, the nodes try to construct the "2x2" shape but it fails and stops in the middle, since "2x2" is an unfeasible solution. It also fails to reach any solutions which are asymmetric with respect to the center of the initial circle. As a result, only a suboptimal ("Cross") solution can be found in this experiment.

2) Initial Circle Center $(x, y) = (840, 920)$ *with Length Restriction:* Refering to the results in Fig. 6, only "L", "1x1"

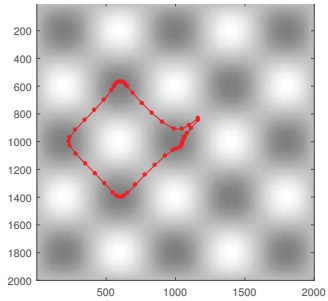


(a) "Cross". $S_K = 464.004$. By taking $R = 150$px. (b) "2x2". $S_K = 464.613$. By taking $R = 500$px.

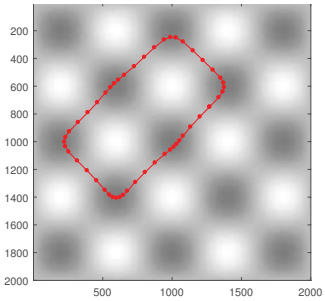
Fig. 5. Types of shapes that can be produced by setting the initial circle's center at $(x, y) = (1000, 1000)$ (center of the map) and varying radius R. Only "2x2" and "cross" (which is a variant of "2x2") can be produced under a symmetric initial condition. "1x1" and "1x2" are asymmetric with respect to paths centered at a peak and thus cannot be produced.



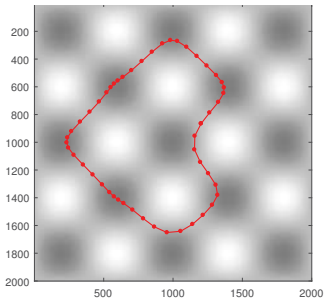
(a) "L". $S_K = 469.736$.
By taking $R = 225$px.



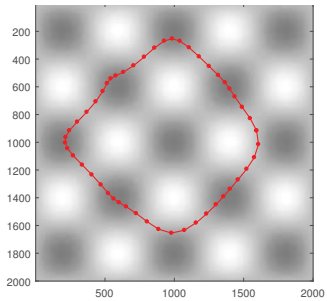
(b) "1x1". $S_K = 469.819$.
By taking $R = 325$px.



(c) "1x2". $S_K = 468.532$.
By taking $R = 425$px.



(d) "3 quarters". $S_K = 464.747$.
By taking $R = 525$px.



(e) "2x2". $S_K = 463.780$.
By taking $R = 625$px.

Fig. 6. Types of shapes that can be produced by setting the initial circle's center at $(x, y) = (840, 920)$ and varying radius R. All the 3 basic shapes and their variants can be produced under an asymmetric initial condition.

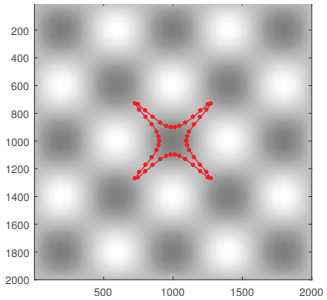
and "1x2" are expected to appear in this experiment since the other two are unfeasible under the given separation constraints. Comparing with the previous experiment, the algorithm performs worse under the more restricted separation constraints. For example, a very regular "1x2" can be generated in the less restricted case as in Fig. 6c, but it is quite distorted under the more restricted case as in Fig. 8c.

We suggest that the distorted shapes are produced because the size of the update are being limited by the separation constraints. The nodes cannot redistribute fast enough along the path. Therefore we can observe that some nodes are overcrowded around the dark regions, while other nodes stretch themselves to the separation limit but are still unable to touch the dark region.

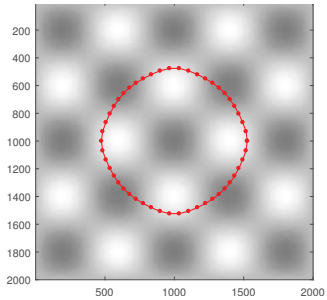
The results indicate that strict separation constraints may cause problems in mission planning, e.g. when the correlation length of the map is much larger than the mean separation between waypoints. The performance is related to the schedule of reducing the parameters α and β in the algorithm, and will be left for further investigation.

C. Search Range Dependency

To demonstrate the dependency on the search range K, we choose the map of population density of US (Fig. 9) as an example. The map is chosen because there is an obvious difference in the correlation length between the East Coast and West Coast regions, which results in different behaviors for different choices of K in our algorithm. In the comparison,

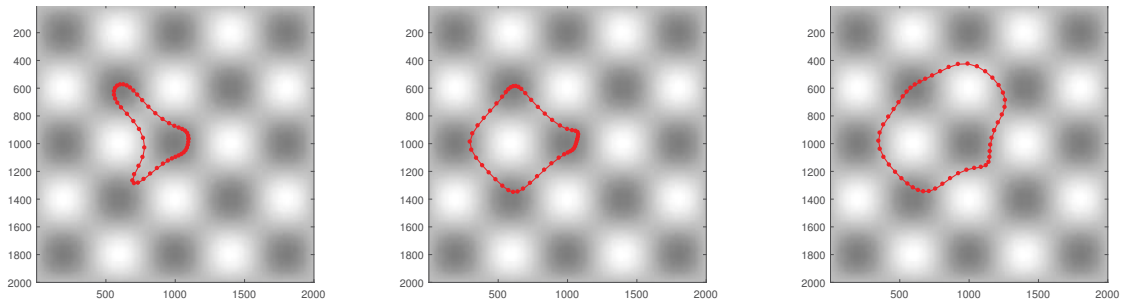


(a) "Cross". $S_K = 459.922$.
By taking $R = 150$px.



(b) Failed "2x2". $S_K = 447.135$.
By taking $R = 500$px.

Fig. 7. Types of shapes that can be produced by setting initial circle's center at $(x, y) = (1000, 1000)$ with restricted separation constraints $[d_{\min}, d_{\max}] = [10, 70]$. Only a diminished "cross" can be formed when the initial radius is small. It also fails to reach "2x2" when large initial radius is used, since "2x2" is unfeasible under the given separation constraints.



(a) "L". $S_K = 463.526$. By taking $R = 225$px.

(b) "1x1". $S_K = 467.721$. By taking $R = 325$px.

(c) "1x2". $S_K = 456.772$. By taking $R = 425$px.

Fig. 8. Types of shapes that can be produced by setting the initial circle's center at $(x, y) = (840, 920)$ with restricted separation constraints $[d_{\min}, d_{\max}] = [10, 70]$. Under this separation constraint, the first 3 cases in Fig. 6 are feasible, but it is observed that only their distorted versions are produced.

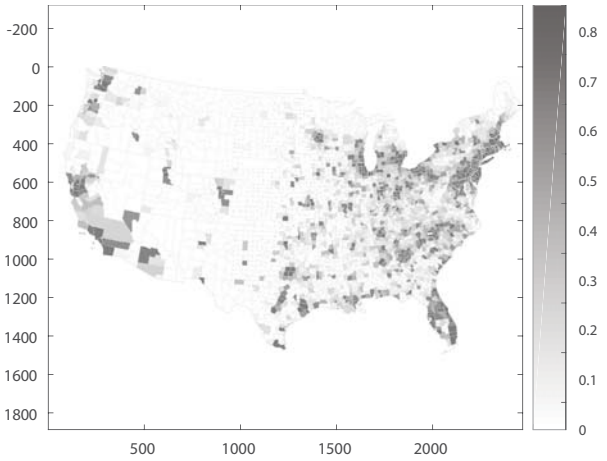


Fig. 9. The US map of population density is chosen to demonstrate how to choose the search range K relative to the map's correlation length. Regions with higher population density are represented by darker colors. The map has a size of $1564 \times 2468\text{px}^2$. There is an obvious difference in correlation lengths between the East Coast and West Coast. *(Source: http://ecpmlangues.u-strasbg.fr/civilization/geography/US-census-maps-demographics.html)*

S_K defined in (1) is not a good indicator since it is proportional to the integration range K. To be fair, we evaluate the the raw score S_0 defined in (2) to evaluate the paths' performances.

In the following experiments, we start with 50 nodes with initial positions as a circle of radius 160px. The separation constraints are set to $[d_{\min}, d_{\max}] = [10, 100]$. In the plots of search range, the yellowish regions represent search range within a radius of K from one of the nodes, while the green regions represent search range within radius of $3K$ from one of the nodes.

1) East Coast: Figure 10 shows simulations with $K = 25, 50, 75$ on the East Coast region. The correlation length at East Coast is small due to the small patch size of the regions ($\approx 20-30$px) and the highly varying distribution. We observe that the path with $K = 25$ gives the best performance in comparing S_0. It is expected because $K = 25$ fits the scale of the correlation length of this area and can do a finer search compared with larger Ks.

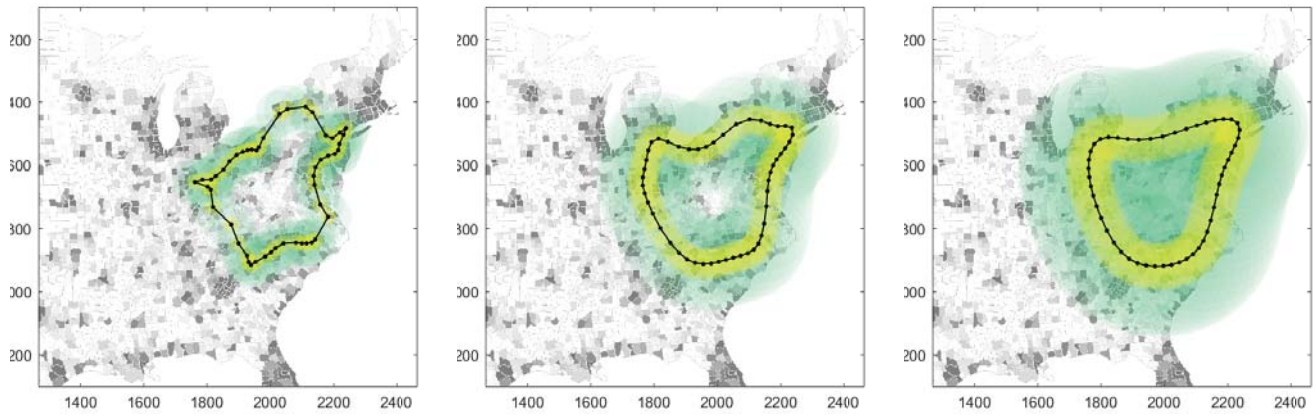
2) West Coast: Figure 11 shows simulations with $K = 25, 50, 75$ on the West Coast region. The distribution in West Coast is sparse, with most population concentrated in California. The area's patch size ($\approx 40-100$px) is also larger compared to that in East Coast. In this case $K = 50$ has the best performance in comparing S_0 because it fits the best the correlation length. We have observed that smaller K such as $K = 25$ fails to refine the resultant path, since the nodes are unable to search for higher density regions with a too small search range. Larger K also performs worse because the search is too coarse.

From the above 2 experiments on the US population map, we suggest that the value of K should be chosen close to the correlation length of the map, so that it can draw a balance between a finer search and large coverage for the best performance.

V. Conclusion

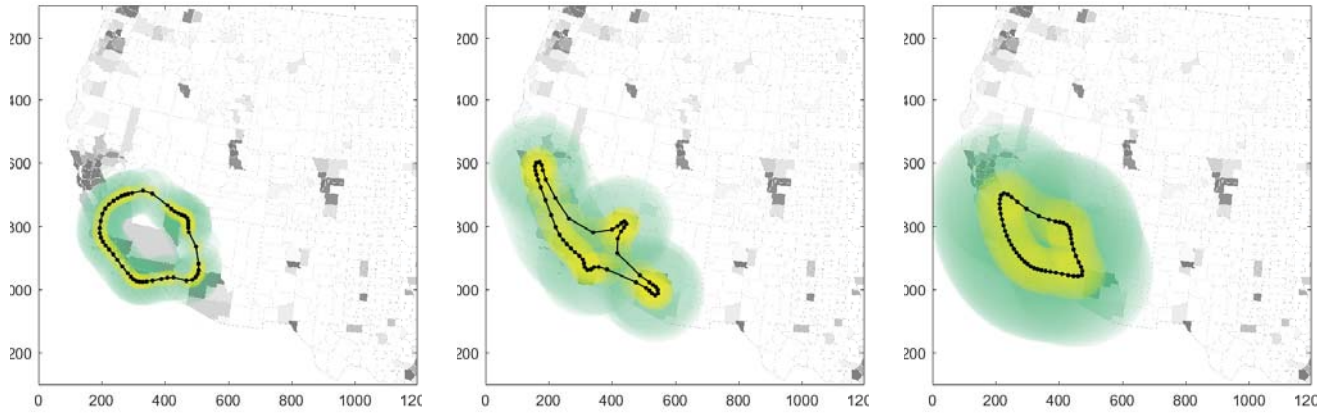
In this paper, we have introduced a new algorithm for generating traverses on a 2D probabilistic map that maximize the integrated probability of finding resources along the path. The algorithm overcomes the limitations of the Elastic Net and adapts to probabilistic landscapes using barrier function constraints. We performed simulations to investigate the performance of the algorithm and its relation to the choice of initial condition, separation constraints of the problem and choice of the search range.

Since the problem is non-convex and multiple optimal solutions exist, a single run of the algorithm is only able to search for a local optimum. In a single run, the test results suggested that the choice of initial conditions is important, and we need to (1) avoid symmetric initial conditions, (2) choose the appropriate separation constraints as far as it is allowed by external factors, and (3) choose K to match the correlation length of the map. In the search for the global optimum, multi-start at different locations of the map is recommended. A redistribution of nodes may also help to improve the results in cases with very strict separation constraints. Taking into account these factors, we obtained excellent results using the US population map as the test bed. For future studies, the optimal schedule of parameter reduction in the barrier method should be considered, and the algorithm can be further improved by optimizing the integrated probability of finding



(a) $K = 25$. $S_K = 370.8099$, $S_0 = 28.0588$. (b) $K = 50$. $S_K = 428.1751$, $S_0 = 16.3961$. (c) $K = 75$. $S_K = 462.3647$, $S_0 = 13.3765$.

Fig. 10. Demonstration on the US East Coast region. The center of the initial circle is at $(x, y) = (219, 1333)$. The resultant paths, range within K (yellow) and $3K$ (green) from the nodes are shown. The normal score S_K, raw score S_0 are also calculated for each path.



(a) $K = 25$. $S_K = 338.6289$, $S_0 = 16.2627$. (b) $K = 50$. $S_K = 423.8759$, $S_0 = 28.8941$. (c) $K = 75$. $S_K = 448.3688$, $S_0 = 20.2118$.

Fig. 11. Demonstration on the US West Coast region. The center of the initial circle is at $(x, y) = (1282, 1037)$. The resultant paths, range within K (yellow) and $3K$ (green) from the nodes are shown. The normal score S_K, raw score S_0 are also calculated for each path.

resources with the overlapping probabilities of neighboring nodes being discounted.

ACKNOWLEDGMENT

This work is supported by grants from the Research Grants Council of Hong Kong (grant numbers 16322616 and 16306817).

REFERENCES

[1] J. Carsten, A. Rankin, D. Ferguson, and A. Stentz, Global planning on the Mars Exploration Rovers: Software integration and surface testing, vol. 26, no. 4, pp. 337357, 2009.

[2] P. Tompkins, A. Stentz, and D. Wettergreen, Global path planning for mars rover exploration, presented at the 2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No. 04TH8720), 2004, vol. 2, pp. 801815.

[3] G. Laporte and S. Martello, The selective travelling salesman problem, vol. 26, no. 23, pp. 193207, 1990.

[4] J. Andrews and J. A. Sethian, Fast marching methods for the continuous traveling salesman problem, vol. 104, no. 4, pp. 11181123, 2007.

[5] P. E. Hart, N. J. Nilsson, and B. Raphael, A Formal Basis for the Heuristic Determination of Minimum Cost Paths, vol. 4, no. 2, pp. 100107, 1968.

[6] Stentz, Anthony, "Optimal and Efficient Path Planning for Partially-Known Environments", Proceedings of the International Conference on Robotics and Automation: 33103317, 1994.

[7] R. Durbin and D. Willshaw, An analogue approach to the travelling salesman problem using an elastic net method, vol. 326, no. 6114, p. 689, 1987.

[8] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2004.

[9] J. Nocedal and S. Wright, Numerical optimization. Springer Science & Business Media, 2006.